User Manual for Multitool

Covering Multitool v1.0b2

Johan Lund
Date: 2005-11-20
Department of Linguistics, Göteborg University
Manual Version: 0.9

Contents

Contents 2

About this Manual 4

What is Multitool? 4

Key Concepts 4

Views 4

Codings 6

Coding Points (CP) 6

Time Points (TP) and Synchronizations 7

Schemes 7

Speakers 9

A/B Points 9

Multitool at a glance 10

MainView 11

Tasks 12

Change the settings 12
Create a new project 12
Creating, Removing, Editing Speakers 14
Store and Recall your Window Layouts 15
Generating the Amplitude file 16
Generating the Spectrogram file(s) 17

Menu Descriptions 21

Buttons 22
File Menu Options 22
Window_Layouts Menu 23
Speakers Menu 23
Views Menu 24

PartitureView 24

Description 24

Tasks 25

Hiding and Showing Rows 25
Creating a new Coding 26
Editing a Coding 27
Deleting a Coding 27
Creating a new Scheme 27
Deleting a Scheme 28
Inserting a CP 29
Deleting a CP 29
Creating a TP 29
Deleting a TP 29
Editing a TP 29
Navigating the Media from the Partiture 30

Splitting a Coding into Two 30

Menu Commands and Buttons 30

Buttons 30
View Menu 31
Coding Menu 31

***TimeScale View* 31**

Tasks 32

Synchronizing a CP 33
Creating a Synchronization from Scratch 33
Creating a Synchronized Coding by Drag-Selecting 34
Creating a Synchronized Coding using A/B points 35
Moving/Fine Tuning Synchronizations 35
Loop Playback 36

Command Descriptions 36

Buttons 37
View 37
Sync 37
Coding 38
Play_length 38
Markers 39

About this Manual

The first few chapters will introduce you to Multitool and make you familiar with some of the key concepts and terminology that will be used throughout this manual. After this we take you through all of Multitool's features view by view. Each chapter that describes a specific view starts with an introduction followed by a "Tasks" section that describes how to perform tasks that are commonly used for that view. Finally, we list a description of what all menu commands and buttons in the interface, for that view, do.

What is Multitool?

Multitool is a cross platform software tool that can be used to analyze and transcribe multimodal communication. With Multitool you can import or create time synchronized, or time stamped, transcriptions using audio and video. Some of its features are;

- Cross platform – Macintosh and PC
- Supports any media format that Quicktime supports
- Amplitude waveform analysis of audio
- Spectrum analysis view of audio
- 6 different views – Partiture, Standard, Audio waveform and Spectrum analysis
- Flexible coding schemes (supports any)
- Time synchronized annotations/codings
- Transcription import formats; MSO6, Tractor, Text
- Colored codings and animation of events when media is playing

Key Concepts

Views

Multitool is made up of different views, sometimes called windows. Each view behaves just like a separate application-window would in your operating system, but they are all connected, "under the hood" so to speak.

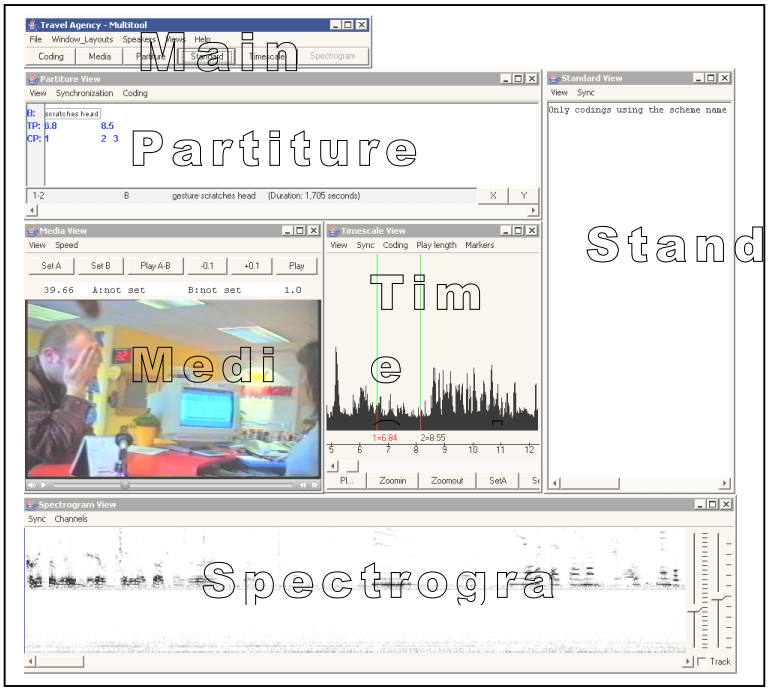


Figure X. Screenshot showing all six views. Observe that the way the different windows or views are laid out may differ.

When you first start Multitool only the "MainView" is shown (unless you have told Multitool to open the last-loaded project upon start, if so; many views might show up). The MainView is fundamentally different from the other views and it is the core of Multitool. If you close the MainView, you will exit Multitool completely, but if you close any of the other views, you just close that view. You could say that the other views are child views of the MainView and depend on it.

There are also some dependencies among the child views themselves. For example; some views display certain aspects of the media such as the TimeScaleView and the SpectrogramView. If there is no media file associated with the project these views can't function. In addition they can not function if the MediaView is closed. So if you want to use the TimeScaleView or the SpectrogramView you must have the MediaView open. Observe that it is possible to open these two views without having the MediaView open; it's just that some of their features will be disabled.

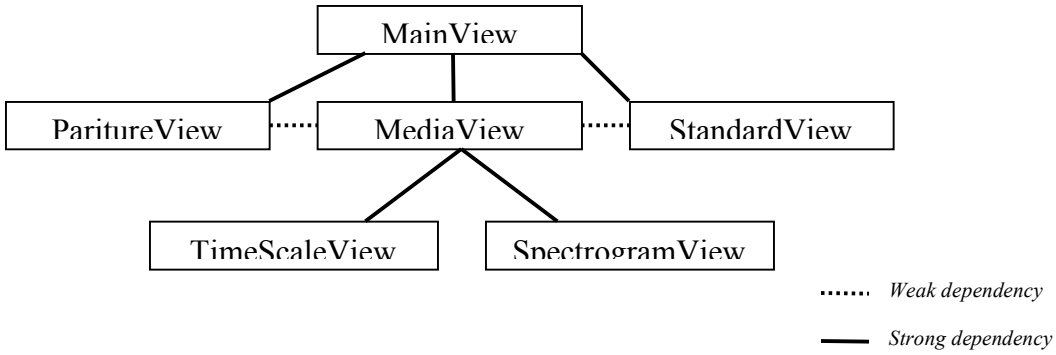


Figure X. Some views need others to be open to function correctly (strong dependency). Some views communicate with- but don't need it (weak dependency).

Codings

In Multitool, there is no real difference between transcription and annotation, i.e. the utterance "no" and the annotation saying that that no is a feedback, are the same thing in Multitool, *codings*.

Coding Points (CP)

Multitool does not work with character positions or time points to order things in sequence, but uses a kind of abstract points called *coding points*. A coding point (CP) is simply a point denoting an interesting event of some kind, e.g. somebody starts talking or an answer ends. All codings are attached to a start CP and an end CP. A coding can also traverse multiple CP's so that it for example; starts at CP 6 and ends at CP 8. CP's are always ordered, ranging from 1 and up, incremented by 1.

<table>
<tr><td colspan="3">Mark walks up to the desk</td><td></td></tr>
<tr><td></td><td colspan="2">Mark scratches his head</td><td></td></tr>
<tr><td>CP 1</td><td>CP 2</td><td>CP 3</td><td>CP 4</td></tr>
</table>

Figure X. Example of how codings and coding points can be used together; at CP 1 mark starts walking up to a desk. At CP 3 Mark has arrived at the desk. The coding "Mark walks up to the desk" then ranges from the conceptual points 1 till 3. Sometime during the mid of his walk, he scratches his head, and as he arrives at the desk he stops doing that. Mark scratched his head between the conceptual points CP 2 and CP 3. Observe that we don't know exactly when in time, these CP's, do occur. We only know the order of things, so far…

Time Points (TP) and Synchronizations

A Time Point (TP) is a point in time, in the media. They are represented with hours, minutes, seconds, milliseconds. A TP cannot exist on its own. It must be attached to a CP. When you create a TP and attach it to a CP you have, as we say, synchronized the CP. We now know where in time, in the media, where the event that the CP represents, occurs. You have created synchronization. Only one TP can be attached to a specific CP at any given time but you can move a TP to a different CP providing that this new synchronization will be in order.

<table>
<tr><td colspan="3">Mark walks up to the desk</td><td></td></tr>
<tr><td></td><td colspan="2">Mark scratches his head</td><td></td></tr>
<tr><td>CP 1</td><td>CP 2</td><td>CP 3</td><td>CP 4</td></tr>
<tr><td>TP 2:21</td><td>TP 2:23</td><td>TP 2:24</td><td></td></tr>
</table>

Figure X. By using TP's we can now synchronize the CP's from the previous example. In this particular movie file that we are annotating, the event, represented by CP 1, occurs at 2 minutes and 21 seconds. It takes Mark three seconds to walk up to the desk at CP 3. During this period Mark scratches his head for exactly one second (rather quickly don't you think) between CP 2 (synchronized at 2:23) and CP 3 (synchronized at 2:24). Observe that CP 4 is empty. You don't have to synchronize CP's, it's optional.

Schemes

A scheme is a category that you assign to a coding. All codings must belong to a scheme of some sort. In Multitool, schemes are created using a free text field. This has both advantages and drawbacks. The advantage is that Multitool can easily accommodate for any set of schemes that you can dream up. The disadvantage is that it demands of you to be very consistent with how you type in text. All schemes that you create can be seen as, being at the top level in a hierarchy. It is however possible to create sub schemes and when doing so it is advisable to use the specific scheme separator sign.

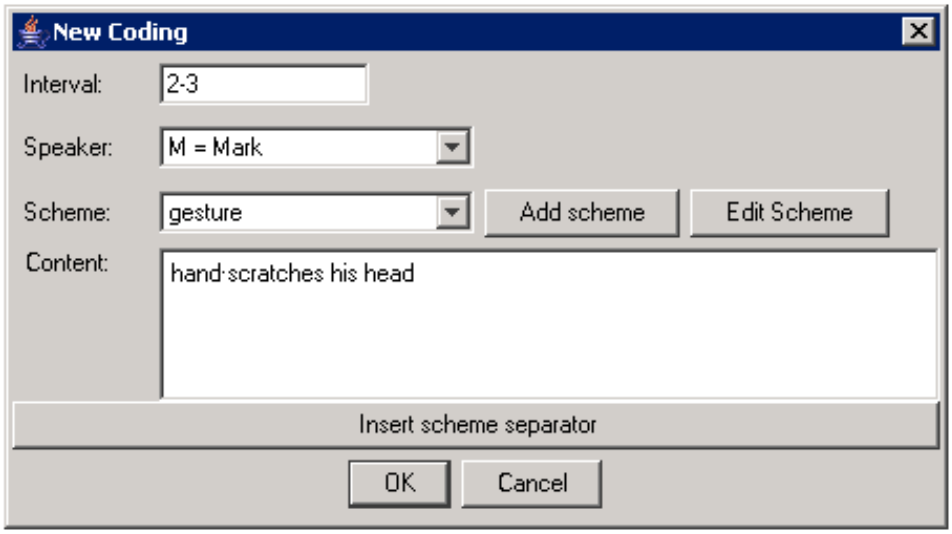


Figure X. This dialog appears when you attempt to create a new coding. Because schemes are stored in a one dimensional list you have to create the sub schemes, such as "hand", by typing them in the content field. You should separate sub schemes (and their content) using the special scheme separator sign. (Notice the little hanging dot between "hand" and "scratches" in the screenshot; it is the scheme separator sign). The advantage of putting sub schemes in the content field is that you will end up with fewer rows in the partiture later on. The drawback is that it calls for you to be more consistent.

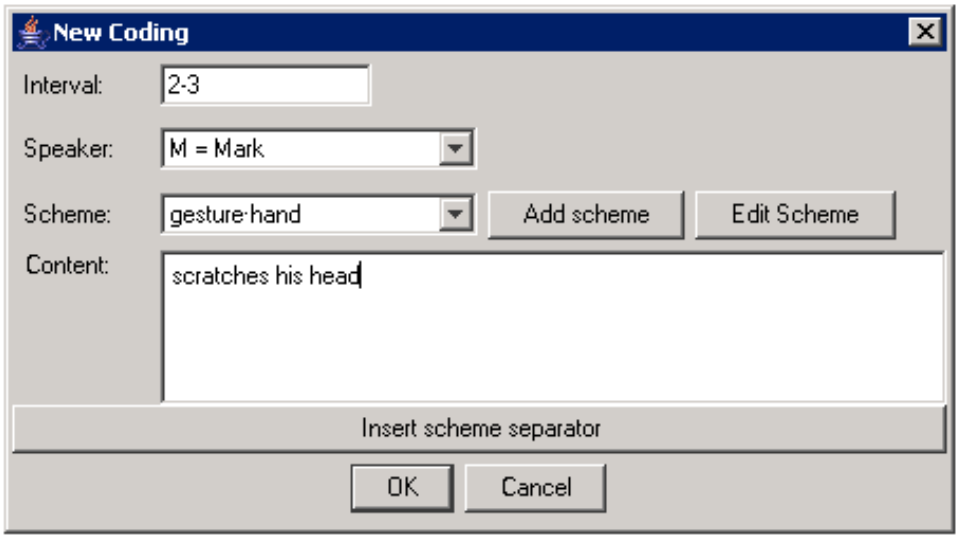


Figure X. Here is an alternative approach. Instead of being stored in the contents field, the sub scheme "hand" has been incorporated into Multitool's list of schemes. (Notice the hanging dot, which is the scheme separator, between "gesture" and "hand" in the scheme box). The advantage of this approach is that you immediately know what schemes you have at hand, looking in the list. You also don't have to type. The drawback is that you may end up with a lot of rows in the partiture later on.

When you create a new scheme you can also define a specific look for it, using colors, font attributes etc., to make the codings that belongs to it, look visually different from codings belonging to other schemes.

It should be noted that there is a special scheme called “text”, which you would have to create if you were to use the StandardView, as only codings belonging to that specific scheme shows up there.

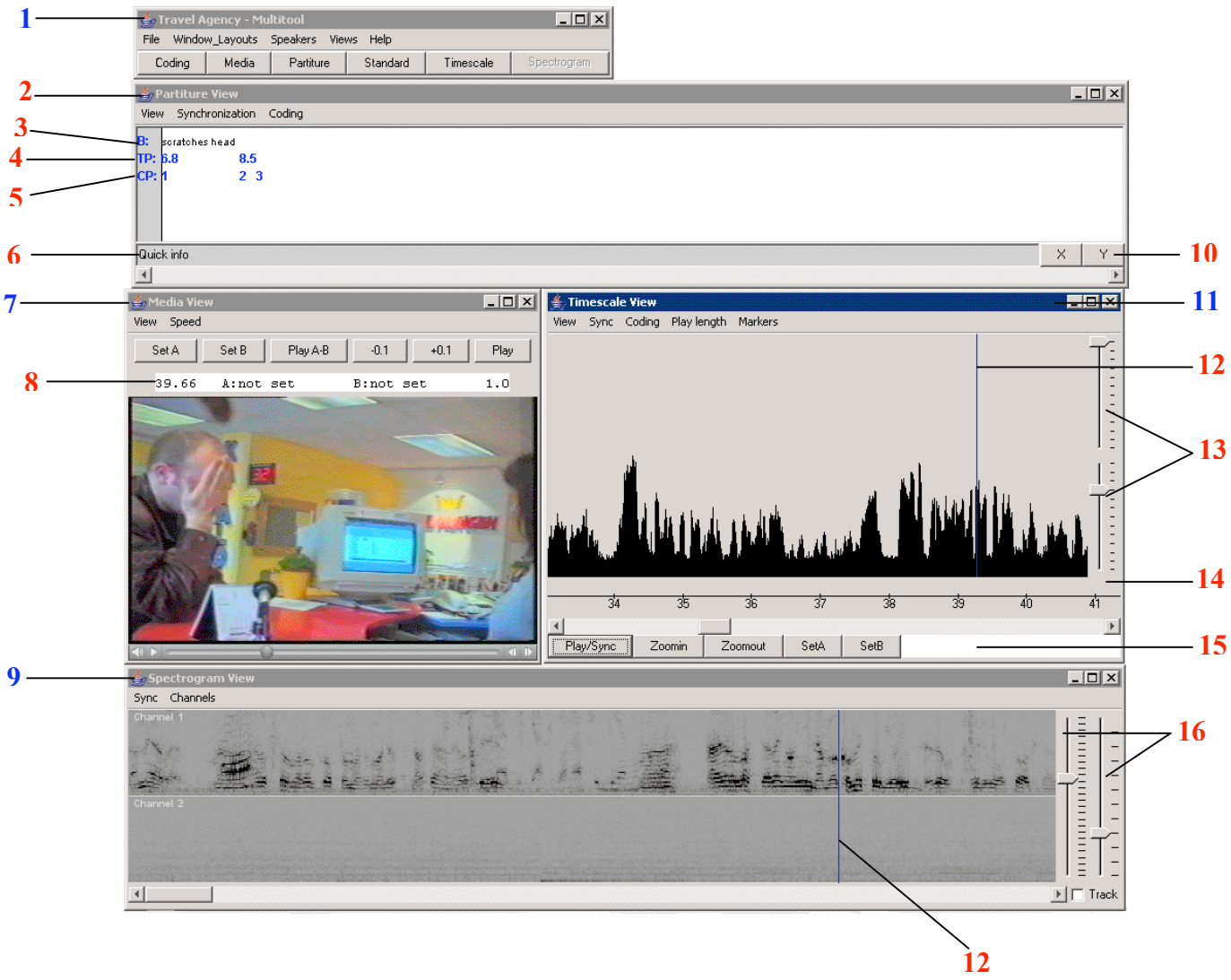
Speakers

The speakers in a Multitool project is simply the people (could be other entities too I suppose) present in the media that you want to annotate or transcribe. A speaker is stored with a full name and his or hers initials.

A/B Points

A/B points can be seen as temporary bookmarks. There is only one A point and only one B point in the whole of Multitool. They fill different roles in different views. For example; in the MediaView a playback loop can be setup by dropping the A point where the loop should start and then dropping the B point where the loop is to end, then pressing the “Play A-B” button. In the TimeScaleView, synchronized codings can be created from the A-point location to the B-point location.

Multitool at a glance



1. Main application window
   Used to create and open projects, open other views, import foreign file formats, speaker management, store window configurations (window layouts), change program and project settings, generate graphics for the TimeScale View and the Spectrogram View.
2. Partiture View
   Shows coded material for each speaker and scheme, shows synchronization information for codings and duration of codings. Used to edit and create new codings, coding points, time points and schemes. Has animated playback that slaved to the MediaView. Can be used to navigate in the media.
3. Speaker initials – one row for each scheme and speaker.
4. Time Points (TP)
5. Coding Points (CP)

6. Quick Info Bar
When the mouse passes over codings in the partiture, this field displays synchronization data and measured duration of coding and scheme of coding.

7. Media View
Displays and allows playback of the media of the project, such as a movie. Enables frame by frame stepping, looped playback, by dropping markers called A/B points, and playback at different speeds as well as media navigation.

8. Media information field
Displays current media location in hours, minutes and seconds. Current position of the A and B points. Current playback speed.

9. Spectrogram View
Shows the FFT spectrum of the audio material in the media (has to be pre-generated). On stereo material; the left (channel 1) and right channel (channel 2) can be dynamically loaded into and out of the computers ram memory. Animated synchronized playback that slaves to the MediaView, can be used to navigate in the media. Can modify the appearance of the spectrum.

10. Grid on/off
Switched the X (horizontal) or Y (vertical) grid on and off in the partiture.

11. TimeScale View
Shows audio as an amplitude waveform. Can be used to; create synchronizations and codings in several different ways, navigate in the media, zoom in and out on the waveform, set A/B points. Animated, synchronized playback that slaves to the MediaView.

12. NowLine
Indicates current media position, and moves as project is playing.

13. Waveform strength mutators
Used to make the waveform higher or lower, one for each channel on stereo tracks.

14. Time scale bar
Shows the time of the media.

15. Synchronization information field
Shows information on the currently selected synchronization, if any.

16. Spectrum mutators
Can be used to brighten the spectrogram, bring out features from the background, reduce noise and invert the colors of the spectrogram.

MainView

Figure X. Screenshot of the MainView.

Tasks

This section describes common tasks that you typically use the MainView for.

Change the settings

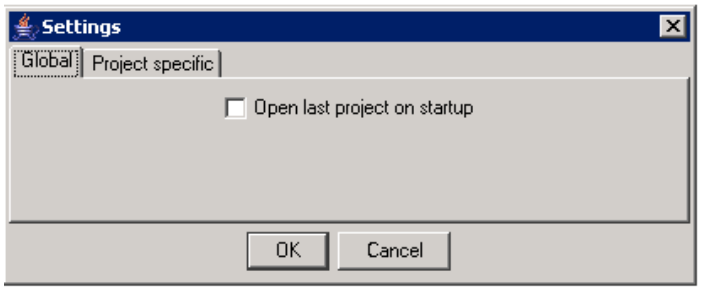


Figure X. The settings dialog.

Global settings are settings for Multitool as a whole while project specific settings are things that affect your current project only. Global settings are saved as XML in a file called MultitoolSettings.xml that is located in the directory that Multitool is launched from. If this file does not exist when Multitool is starting up, it will be created with default settings. Project specific settings are stored with the current project in a file called <projectname>.mtp.

Create a new project

This is done by going to the *File* menu and selecting *New*. A dialog appears.

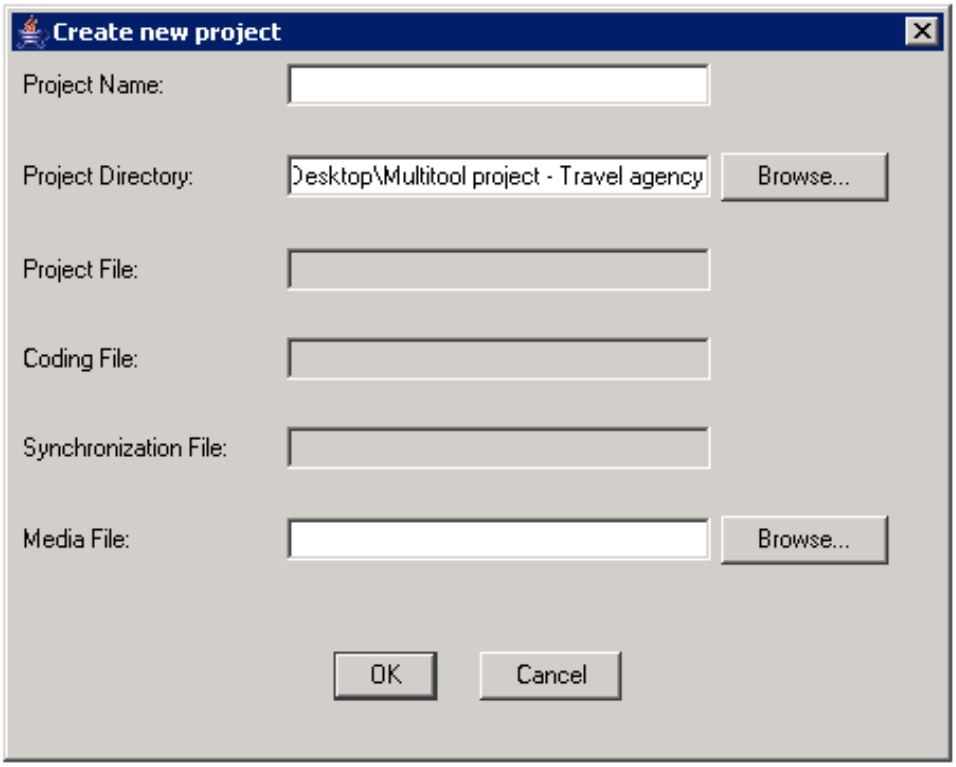


Figure X. The new project dialog.

A Multitool project actually consists of several different files, some of which will be automatically generated now. A requirement is that all these files reside in the same directory/folder on your computer. It is therefore a good idea to, in advance to creating a new project, place the media file in the directory where you intend to keep the new Multitool project.

Start by typing in a descriptive name for the project in the first white box. The second white box, from the top, is where you state what directory/folder on your computer you want the project to be generated. You can type in the full path manually but it is much easier to browse for the intended location by pressing the browse button to the right.

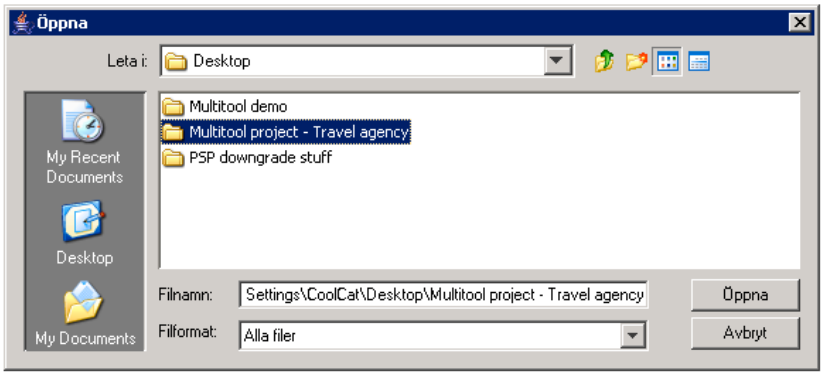


Figure X. The browse for directory dialog box (PC version / Swedish).

This opens the *browse for directory/file* dialog box. Select the destination directory for the project, by single clicking it (double clicking it takes you into the directory) and then click the *open* button. The browse dialog will close and take you back to the *new project* dialog where the full directory path will be shown in the *project directory* box.

Project file, coding file and synchronization file boxes are grayed out and cannot be changed. This is because Multitool will create, and name them for you, using the project name as a base. All files will be placed in the directory you have specified.

Finally, you specify what media file you want to use. It is not mandatory to have a media file associated with the project but you will only be able to use the partiture view and the standard view if you don't have one. Press *browse* next to the white box and select the media file. (The media file must be present in your *project directory* or you will receive an error message). Press *OK* to close the dialog.

Multitool now suggests that you create some speakers, by showing you the *create speakers* dialog. You will need at least one speaker to be able to do any work.

Creating, Removing, Editing Speakers

A project needs at least one speaker in order for you to be able to code. The *Edit Speakers* dialog can be accessed from the *Speakers* menu. When you create a new project it will also pop up automatically. It presents a list of all speakers available in the project. Speakers are shown with their full name and their initials surrounded by parentheses.

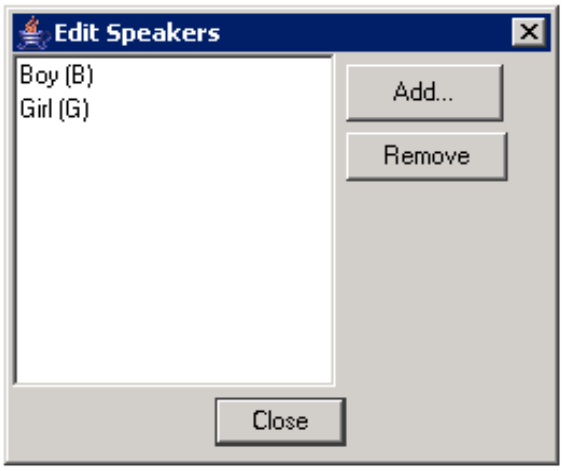


Figure X. The edit speakers dialog.

To close this dialog just hit enter or press the *close* button.

To delete a speaker; click on it in the list and then press *Remove.* You will be asked to confirm. Be very careful when you delete speakers that have codings attached to them as these codings will be lost too.

To create a new speaker; click add. The *Add a Speaker* dialog appears. Type in the information and hit enter or press ok.

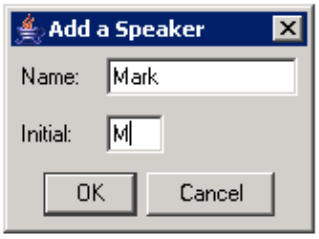


Figure X. The dialog that lets you create new speakers.

Store and Recall your Window Layouts

All the windows that make up Multitool can be arranged in many different ways to tailor to specific needs. The size and location of all open windows are stored with the project but there are times when you need to rearrange the interface. If you for example; receive projects from colleagues etc. whose window layouts differ from what you are used to, it can be quite painstaking to constantly resize and position them. Multitool solves this by letting you store various window layouts in a global repository (MultitoolSettings.xml) and later recall them. This immediately reorganizes the windows on the screen acoording to your specification.

To save a window layout set; open the views you commonly use and lay them out on the screen resizing them as you wish. From the *window_layouts* menu, choose *store current layout.* You will be asked to name this window layout.

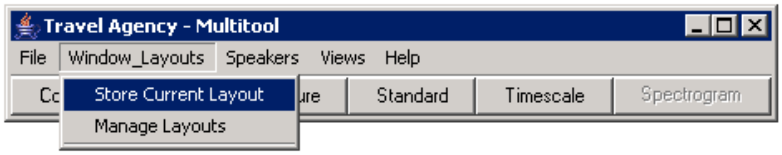


Figure X. This menu option saves the position and size of all currently opened views.

To recall a window layout set; you go into the *Window_Layouts* menu again. You will find that you have a new entry with the name you specified. Select it and all your views will be restored on the fly to the positions and sizes they were when the layout was stored.

Figure X. This menu option recalls the size and positions of all views in a previously stored window layout.

To delete layouts you will have to go into the *Manage Layouts* dialog.

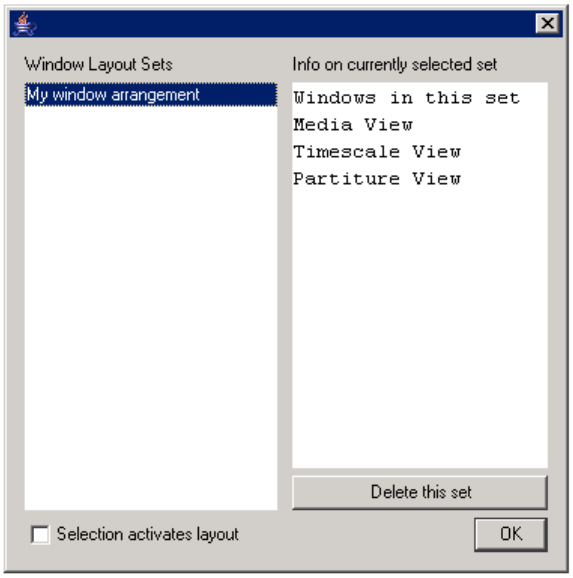


Figure X. The Manage Layouts dialog can be used to check out previously stored layouts as well as delete them.

Stored layouts appear on the left hand side. You can left click layouts to select them; this also fills the right hand text box with information on what windows were opened when the layout was stored. Clicking the *Delete this set* button deletes the selected layout. The checkbox S*election activates layout* causes layouts to activate and windows to fly around the screen as you select different layouts in the right hand list.

Limitations of window layout sets

- Only size and position of windows is stored.
- Only windows that were open when a specific layout was stored can be manipulated by the window layout manager. (You can see which views/windows that will be affected in the *Manage Layouts* dialog).
- Changing your monitor's resolution might cause undesired results. (Just make some new layout sets).

Generating the Amplitude file

This is something you will have to do before you can use the TimeScaleView.
To do this, go *File->Generate amplitude file.* A file called <projectname>.mta will be generated in the project directory. If the .mta file already exists, you will be asked to confirm to overwrite it. This process is cpu intensive and might take some time to complete, especially if you are working on a very long video. If the TimeScale view is already open for some reason, it is a good idea to close it first.

Unfortunately this does not work with all media files. You may find that the MediaView can play back the file fine, but the amplitude waveform can not be calculated. In order to explain why this happens I will have to talk a little about what goes on under the hood. It's mostly technical information so now you have been warned…

Generating the amplitude file (.mta) can be broken down into the following steps:

1a. Try to extract audio from video using QuickTime.
(Audio is stored temporarily in a file).
1b. Try to extract audio from video using JMF.
2. Read the temporary audio file and construct an intermediate form file (the .mta).

Now here is the trick; QuickTime comes in two flavors namely *standard* and *pro*. Both versions are really good at playing back many different kinds of media. But if you want to convert media between different formats, which is what we're doing in step 1a, you need to have QuickTime Pro. So for most people 1a won't work. We then try extracting audio from video using JMF but JMF can only do this if the file is of type Mpeg1 muxed (also called interleaved). And if the file is not of that kind we're not going to get to stage 2. The problem can often be remedied by upgrading to QuickTime Pro. For some bizarre reason the MPEG1 muxed format is one of the few formats that QuickTime Pro cannot convert. It is still also one of the most common formats around. That is why I have left the JMF code in. What can we learn from this?

If you don't have QuickTime Pro; stick with mpeg 1 muxed video files.

Generating the Spectrogram file(s)

This must be done in order to enable the SpectrogramView. To generate it use; *File->Generate Spectrum File(s)*. If the sound of the media is mono then one file called <projectname>.sp1 will be generated in the project directory. If the sound is in stereo an additional file <projectname>.sp2 will be generated also. If the SpectrogramView is already open, Multitool will automatically close it and free up memory.

When you initiate the command, Multitool will first extract a temporary audio file (using JMF only, see the discussion in the amplitude generation section). You will see a small progressmeter telling you it is extracting audio. This audio will later on be fed through an FFT analysis algorithm that will produce the two intermediate files (.sp1 and .sp2). There are many settings for the FFT algorithm that will produce different looking spectrograms. A dialog will show up urging you to select settings for the FFT.

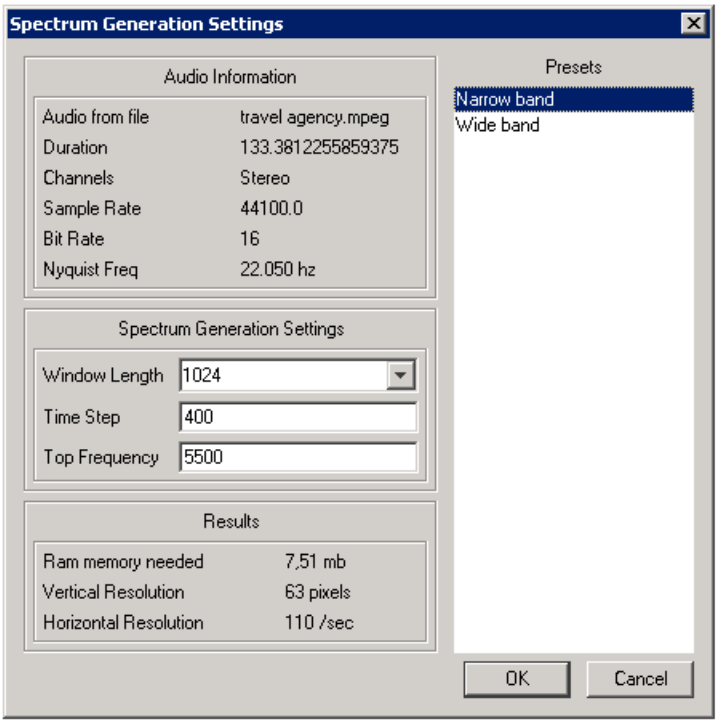


Figure X. FFT analysis settings dialog.

The dialog is divided into four sections.

Audio information
This section displays information on the audio file that is to be analyzed (a temporary file extracted from the project's media file). The duration of the file is displayed in seconds etc.

S*pectrum generation settings*
This section lets you manually enter values that will affect the outcome of the generation.

Window Length
Size of data chunks read, measured in samples. It influences frequency resolution and it must be a value that is a power of 2. Some predefined values have been loaded into the list.

Time Step
Size of the gap between two data chunks start positions. It influences temporal resolution. If you want to know this value measured in seconds instead use this formula:
Time step / sample rate

Top Frequency
This option enable you to cuts away all frequencies above this frequency in the final FFT data.

For an in depth explanation of what the parameters means see below.

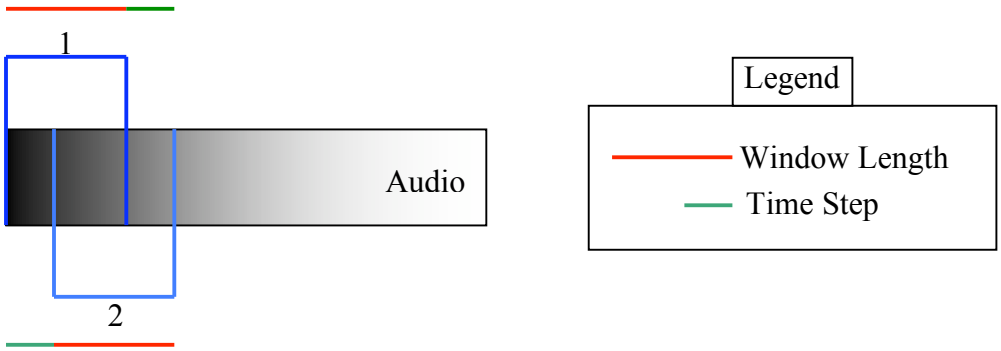


Figure X. This illustration shows that data is read in chunks (blue), from an audio file. The size of the chunks is measured in samples and is called "window length" (red). When the next chunk (number 2) is read it steps forward just a little bit, overlapping the data that chunk 1 just read. The size of this step is called "time step" (green).

So how does *window length* and *time step* affect how our spectrogram looks?
Each chunk is fed through an FFT algorithm that performs some number crunching on it. But the FFT swallows some of the numbers in the process. When the chunk comes out of the FFT it is only half the size and only represents half of the frequencies (nyquist frequency). So if we have a window length of 1024 samples we get back 512. These 512 are now actually frequency strengths logarithmically sampled across the original data's nyquist frequency (half the original sample rate). So in a given moment in time (a chunk is like a snapshot) we have 512 frequency strengths. In a spectrogram frequencies are represented vertically, and time is represented horizontally. So if we want to see all frequencies between 0 and nyquist frequency, painting one frequency with one pixel, we get a maximum vertical resolution of 512 pixels.

When it comes to looking at speech however, frequencies above 5.5k are not that interesting, that is why we have the option to cut all frequencies above a certain threshold with the *Top Frequency* option in the *Spectrum Generation* dialog. This will further lower the height, in pixels, of the spectrogram. It will save us some ram memory though. To counteract and give us some more pixels to play with in Y-led we could increase the Window Length. The FFT process is an average of the data in the chunk, so having too large chunks will cause the data to look less well defined.

Is *Time step* then the temporal resolution? Yes. The lower the time step the more chunks get read from the same data. It is however not always necessary to strive for high resolution. Too much data can lead to a cluttered image and consumes more ram memory for example.

What have we learned so far?

- *Window Length* is vertical/frequency resolution
- Too low *window length* causes very few pixels representing data in Y-led.
- Increased window length gives better frequency resolution but blurs it more.

- *Time Step* is temporal resolution
- A high *time step leads to lower horizontal/temporal resolution but makes it less cluttered and consumes less ram.*
- A low time step increases temporal resolution but creates more overlap between the chunks, possibly causing noise/blur.
- The *top frequency* parameter cuts out unwanted frequencies, saving ram, possibly forcing us to increase window length to counteract, leading to more noise/blur.

The four spectrograms shown below, all display the same utterance but using different settings.

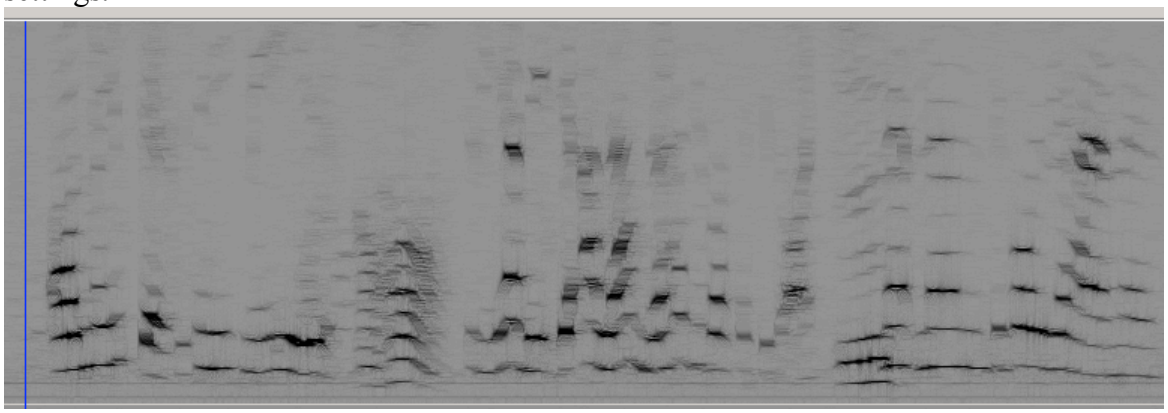


Figure X. Spectrogram with settings: window length=4096, time step=400, top frequency = 5500.

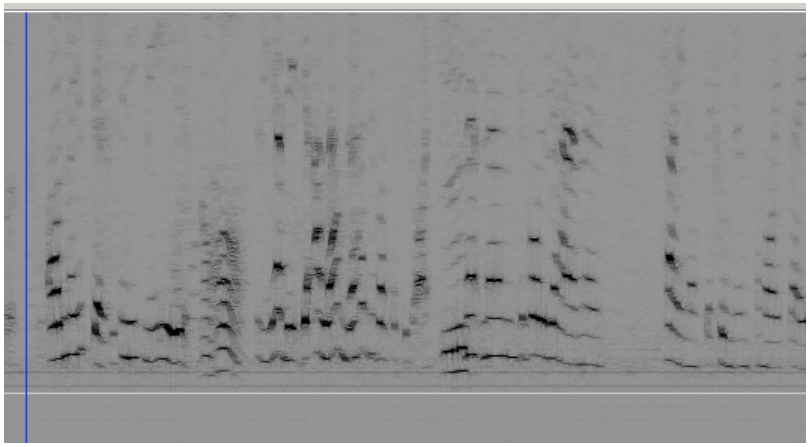


Figure X. Spectrogram settings: window length=4096, time step=800, top freq=5500.

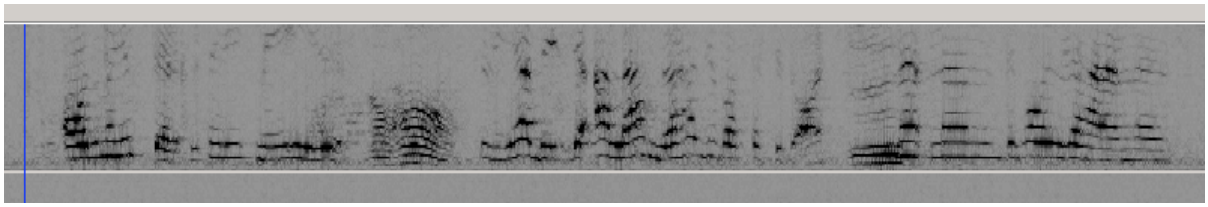


Figure X. Spectrogram settings: window length=1024, time step=400, top freq=5500.

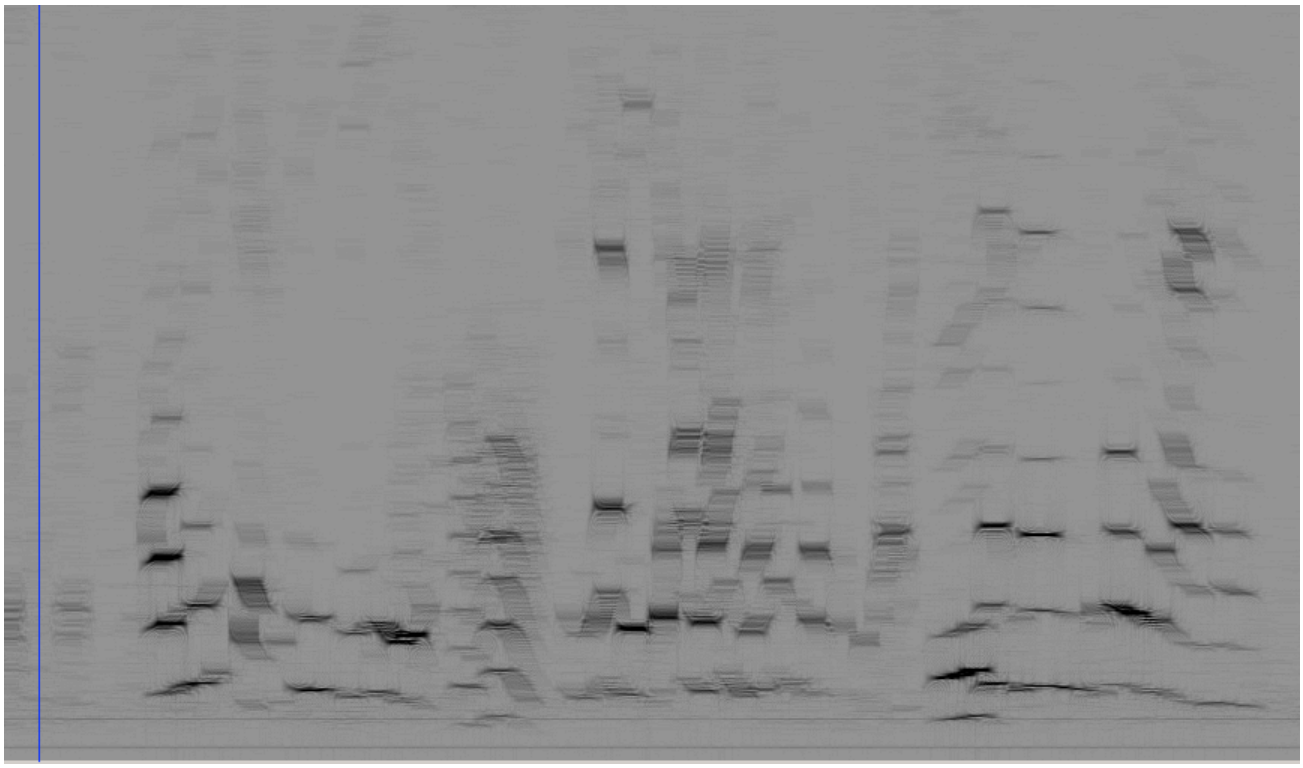


Figure X. Spectrogram settings: window length=8192, time step=400, top freq=5500.

Ram memory needed
The data for the whole spectrogram must be resident in the computers ram memory. This text field displays how much ram the spectrogram will consume, based on the current parameters. If the media is stereo you can later on, in the spectrogram view, disable one of the channels to save half of the memory. Pay attention to this value for large files. You don't want this value to start approaching your physical ram.

Vertical Resolution
How many pixels in Y-led we can expect the data to be represented by, based on current settings.

Horizontal Resolution
This is also called temporal resolution. Indicates how many pixels per second there will be, representing the data, based on current settings.

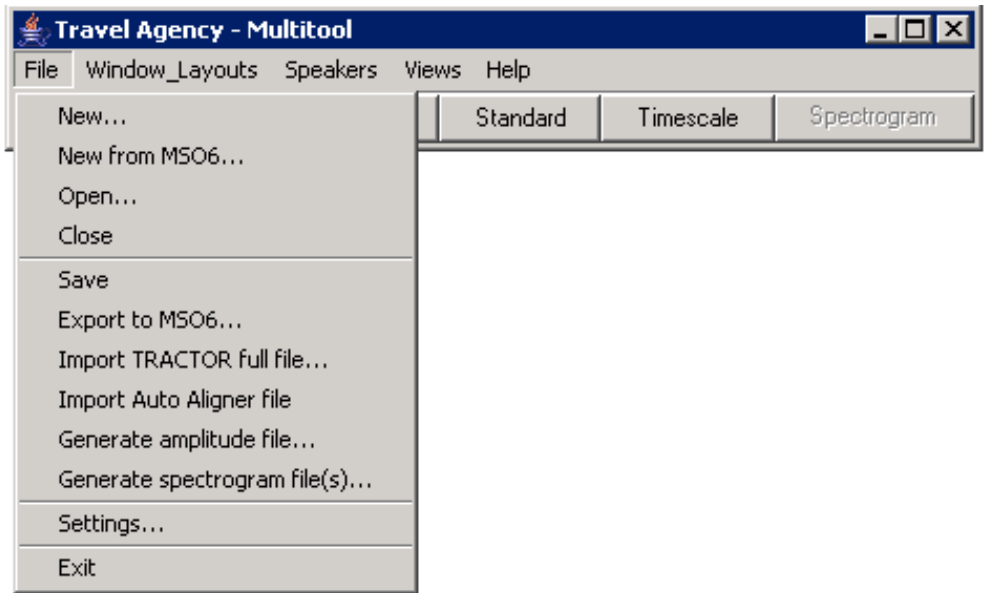
Presets
Clicking one of the presets in the list loads predefined values into the *Spectrum Generation Settings* boxes. It is not currently possible to save your own presets.

Menu Descriptions

Buttons

The buttons Coding, Media, Partiture, Standard, Timescale and Spectrogram launches the corresponding view. Observe; the Coding view is currently not implemented. To launch the TimeScaleView the amplitude file would have to be generated first. To launch the spectrogram view the spectrum file(s) would have to be generated first. Otherwise the option may appear grayed out.

File Menu Options



New
Creates a new Multitool project.

New from MSO6
Starts a new Multitool project by importing an existing transcription from an MS06 file.

Open
Loads an existing Multitool project from disc.

Save
Saves (and overwrites) the currently open project.
Note: There is no “Save As” command in Multitool. To make a backup copy you will have to copy the files, which constitute the project, to a safe place manually.

Export to MSO6
Exports the contents of the partiture to an MSO6 compatible file.

Import TRACTOR full file
Import a tractor project.

Import Auto Aligner file
The auto aligner (AA) is an experimental tool that automatically synchronizes a transcription, using AI techniques. To import data generated by the AA you first create a new project with the relevant media file and then use this option.

Generate amplitude file
This must be done before the TimeScaleView can be used. It scans the audio inside the media and generates a graphical representation, based on amplitude, of it and stores the result in a project specific file called <projectname>.mta.

Generate Spectrogram File(s)
This must be done before the SpectrogramView can be used. It scans the audio inside the media and calculates the FFT of it. The result of the computation is then stored in either one file (for mono media) or two files (for stereo media) called <projectname>.sp1/sp2.

Settings
Used to modify the behavior and appearance of Multitool or your currently loaded project.

Exit
Closes all views and exits Multitool. You will be asked if you want to save before.

Window_Layouts Menu

Stores window layout sets in a global repository. All Layouts you create will be listed under this menu for easy access.

Store current layout
Captures the size and location of all currently open views in a so called “window layout set”, that can later be recalled to save you from the work of moving and resizing windows a lot.

Manage layouts
Opens a dialog where you can delete, step through to test layouts and see information about layouts.

Speakers Menu

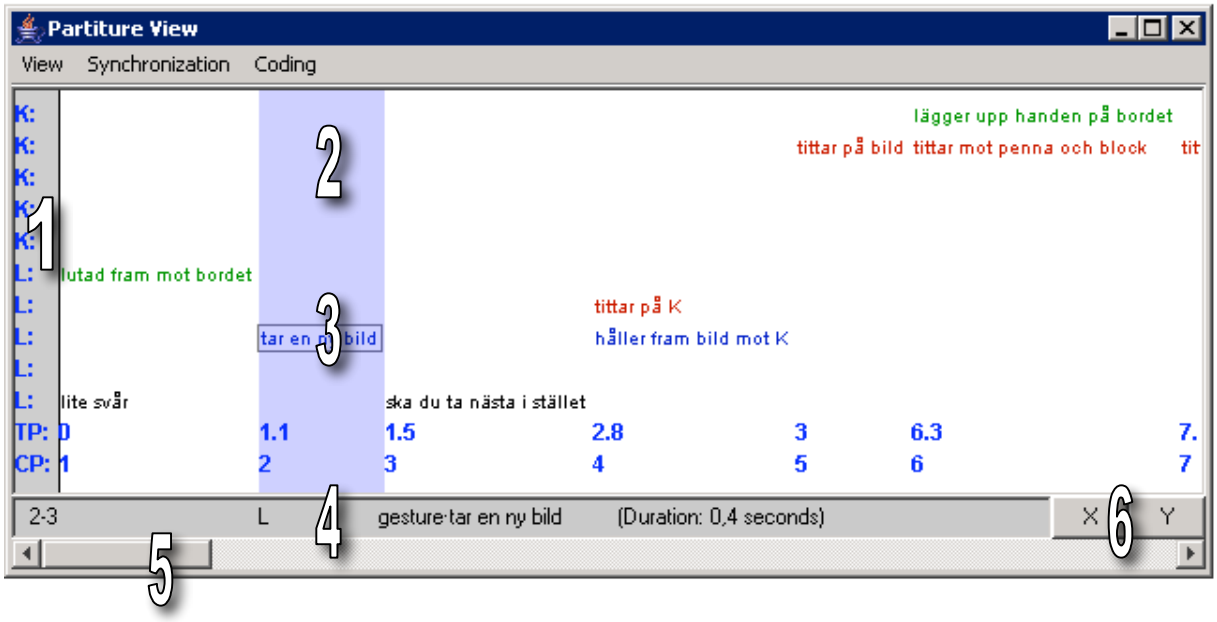
Edit speakers
Brings up a dialog that lets you create and delete speakers.

Views Menu

Same functionality as the buttons but with the addition of a *close all* command.

PartitureView

Description



1. *Rows*
   This gray field called “row headings” shows the initials for the speakers in the project. Note that there are several row’s for each speaker in this partiture. The reason is that there are several different schemes available and each speaker gets a row for each scheme. For example; we have one speaker (K) and two schemes (1 and 2). This creates two rows (K1 and K2). If we add another speaker (L) we now have four rows (K1, K2, L1, L2). However, if there are no codings for say L2. Then the row for L2 will not show up in the partiture.

   There are two special rows at the bottom at the partiture. The first one is the TimePoint (TP) row. It’s time points, if any, shows up right above the CodingPoint (CP) row’s coding points indicating that they are synchronized points.

2. *Now Line/Field*
   This blue field moves along the partiture when the media is playing. It tells you which part in the partiture/media is currently playing. This field can only be drawn where there is synchronized codings. If, for example in the above

screenshot; CP 3 did not have a TP connected to it, the field would have been drawn from CP 2 to CP 4. If we are at the last synchronized CP in the project, the now field will not draw at all. So synchronize your codings and you'll know where you are.

3. *Focused codings*
   This coding has a bounding rectangle surrounding it. It means that the mouse have just passed over it.

4. *QuickInfo bar*
   This field displays information about the currently focused coding, if any. The information you see is explained in order of appearance from left to right.
   The coding's start and end CP's (2-3). The initials of the speaker for the coding (L). The scheme and content of the coding (gesture…). The duration, measured in seconds, of the coding (its start and end points must be synchronized in order to calculate this value. The QuickInfo bar is therefore also useful if you want to find out what scheme a certain row represents. Just hover over any coding on that row.

5. *Scrollbar*
   Drag this to see the rest of the project.

6. *Grid toggle*
   The X button activates a horizontal grid with lines tracing the contour of the rows. The Y button activates a vertical grid with lines drawn at every coding point.

Tasks

Hiding and Showing Rows

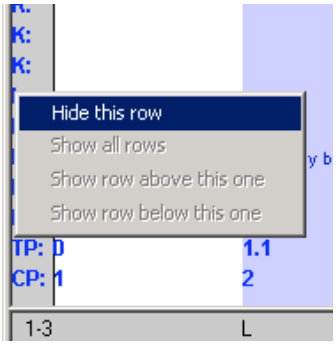


Figure X. Popup menu that appears when right clicking on a row header.

Hiding a Row
It is possible to hide individual rows by right clicking on the heading of a row and selecting the option to *Hide this row* from the popup menu that appears.

Showing hidden rows again
Once you have hidden one or more rows, the option to show all rows becomes available on the popup menu.

Creating a new Coding

Method 1
In the *Coding* menu, select the option *New,* to make a new coding. A dialog appears.

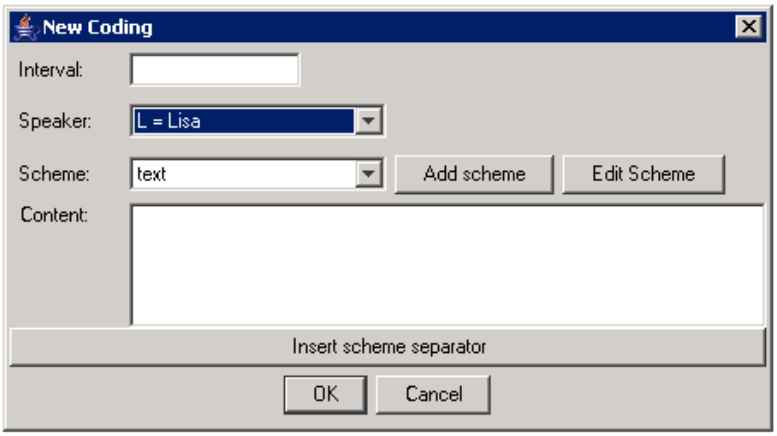


Figure X. New Coding dialog.

Interval
Should be a string with from and to coding points (1-2). The coding can also span several coding points (e.g. 1-6). The coding points you type in must already exist.

Speaker
Select a speaker from the list that this coding concerns. If you need to create additional speaker you must turn to the MainView.

Scheme
Either select an already existing scheme from the list or add a new scheme to the list by pressing *Add scheme.* The appearance of schemes can be changed at any time with *Edit Scheme.*

Content
Transcribe/annotate here. The field is a multi line field but only one line will display in the partiture.

Insert scheme separator
Place the cursor in the *content* field and click this button to insert a “hanging dot”. This character might be used in a future version to separate schemes.

Method 2

In the partiture; right click on a CP and select the option *"Create coding starting at this CP"*. The *New coding* dialog will now appear with the interval filled in for you.

Figure X. This is what happens when you right click on a CP.

Method 3
Instead of using the popup menu on the CP you can also just double click the CP.

Editing a Coding

Simply right click on a coding and select the option to *Edit this coding.* You can also just double click the coding.

Figure X. This is what happens when you right click on a coding.

Deleting a Coding

Method 1
Right click on the coding in the partiture and select the option *Delete this coding.*

Method 2
Hold down the shift key and double click on the coding.

Method 3
Edit the coding, empty the *Content* field and press *Ok.* You will be asked to confirm the deletion of this coding.

Observe, that if the coding that you delete is the last one using that scheme, that scheme will no longer be in use in the project and will promptly be deleted.

Creating a new Scheme

To make a scheme we have to "go through" the *New Coding dialog*. There is currently no other way to get to the button that creates schemes for us. In the *Coding* menu, select the option *New,* to make a new coding. A dialog appears.

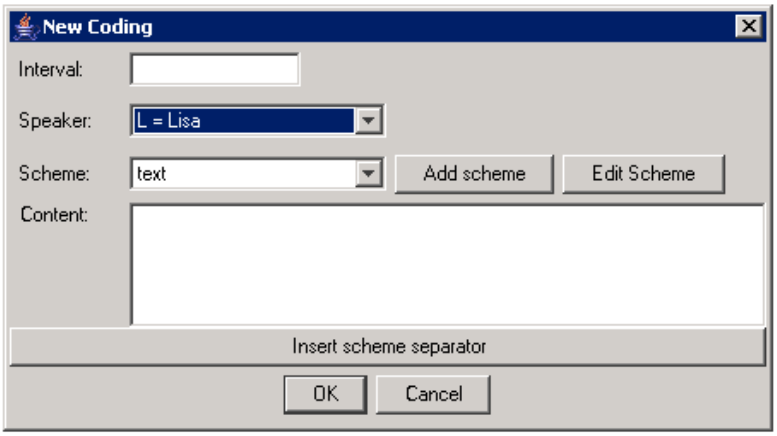


Figure X. The "New Codig" dialog must be used in order to access the Add/Edit Scheme buttons.

Press *Add scheme* and a dialog called *Create new scheme* will appear.

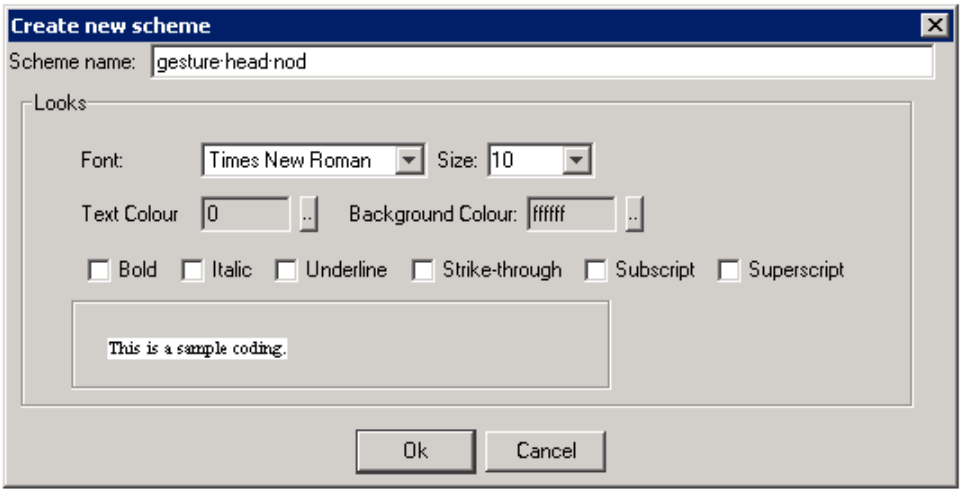


Figure X. The create new scheme dialog.

Type in a descriptive name in the *Scheme name* text box at the top. Use scheme separators if you want (remember our discussion from the key concepts chapter). You can change the appearance of the scheme and a sample of how it will look is displayed in the lower section. Press *Ok* to finish.

You edit an existing scheme using the same approach, only clicking *Edit scheme* instead of *Add scheme.* You cannot however change the name of the scheme after it has been created. The *Scheme name* box will be grayed out in that dialog.

Deleting a Scheme

When there are no longer any codings, for any speaker, which uses the scheme it will be removed automatically. Observe that if you don't know about this, it may seem like schemes mysteriously disappears.

Inserting a CP

Double click in the white on the CP row. A new CP will be interpolated between existing CP's if needed.

Deleting a CP

You cannot delete a single CP of your choice. You can only tell Multitool to remove all unused CP's from the partiture. You do this using the *Purge* command from the *Coding* menu.

Creating a TP

Creating TP's is much easier using the TimeScaleView but if you really want to you can go into the *Synchronization* menu and choose *New.*

Deleting a TP

Method 1
Hold down shift on the keyboard and double click the TP.

Method 2
Right click on the TP and select the option *Delete this TP.*

Method 3
Double click the TP so the *Edit TimePoint* dialog pops up. Click *Delete.*

Editing a TP

Method 1
Double click the TP. Make your modifications in the dialog that appears and click update.

Method 2
Right click on the TP and select the option *Edit this timepoint.* When done click *Update.*

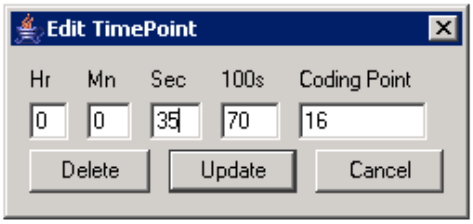


Figure X. Dialog used for manipulating existing TimePoints.

Navigating the Media from the Partiture

Moving the Media to a TP
Single left click on any TP and the movie will position itself at that time. The MediaView must be opened for this to work as the PartitureView sends media position requests to it.

Start/Stop playback
Press space bar to start playing from the last known media position. Doing this from the partiture bypasses loops and A/B points set in other views.

Splitting a Coding into Two

Move the mouse cursor inside the coding that you want to divide. Press down and hold the left mouse button at the character position where you want the split to occur, and then drag the mouse to, either left or right. A selection starts forming and eventually snaps to the end of the coding. Now go to the *Coding* menu and choose the option *Split coding.*

Figure X. A selection has been made inside a coding. The coding is now ready to be split apart.

Caution: Please use this command with caution. We have not decided on the exact logic of it yet. Picture this:

Split Here

Coding A (1-4)			
	Coding B (2-3)		
CP 1	CP 2	CP 3	CP 4

The problem is; if we split coding A somewhere, what do we do with coding B? should it also be split or put completely before or after coding A?
Try using the split function, if you need it, to get an understanding of how it is currently implemented.

Menu Commands and Buttons

Buttons

X
Activates a visual horizontal grid tracing each row

Y
Activates a visual vertical grid tracing each CP

View Menu

Play/Stop
Toggles media playback on and off

Close
Closes the partiture view

Synchronization Menu

New
Creates a new time point

Coding Menu

New
Lets you create a new coding. It opens the *new coding* dialog with all fields empty.

Purge
Detects and removes unused coding points. Coding points are considered unused if they don't have a coding starting at it, ending at it or passing over it and no time point is attached to it.

Edit Speakers
Brings up the *edit speakers* dialog letting you add/edit/remove speakers.

Split Coding
Lets you divide a long coding into two shorter codings. A text selection must be made inside a coding before this command works.

TimeScale View

This view can be used for a number of things such as navigating the project, creating synchronizations and synchronized codings, and setup play loops etc.

There are a few things that you need to be aware of before you start using the TimeScale view. First; it relies on the MediaView which must be open and secondly you need to generate the amplitude waveform from the MainView first. If you are going to create codings from the timescale view you might need to have the partiture open too.

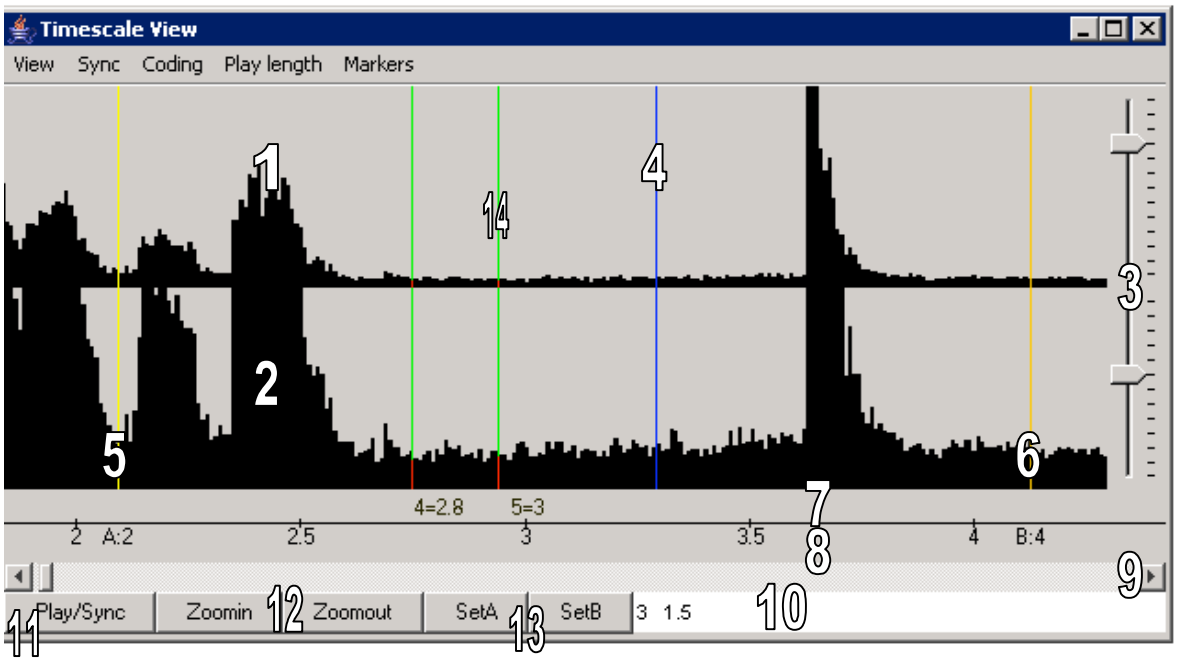


Figure X. The TimeScale view and it's various features.

1. Amplitude strength of left audio, left channel.
2. Amplitude strength of left audio, right channel.
3. Amplitude strength compensation sliders. Used to increase decrease the height of the amplitude waveform.
4. Now line. Indicates current media play position.
5. Current *A Point* location (bright yellow).
   (A:2 underneath means point A at 2 seconds)
6. Current *B Point* location (dark yellow).
   (B:4 underneath means point B at 4 seconds)
7. Synchronized coding points (CP=TP). One for each green line.
8. Media time scale and A/B points.
9. Scrollbar that can be used to navigate in the media. (Locked when media is playing).
10. Information about currently selected synchronization (displayed as red line but out of view in this screenshot)
11. Toggles media play on and off.
12. Zooms in and out on the amplitude waveform display.
13. Buttons that can be used to drop the A/B markers at current play position.
14. A green line indicating synchronization.

Tasks

Synchronizing a CP

To synchronize a CP is to connect it to a point in time (a TP). We assume that there is an unsynchronized CP already in existence. (You can insert new unsynchronized/free CPs from the partiture). Place the blue "now line" where you want the new sync to be. From the menu; choose S*ync->New* and the "Synchronize CP" dialog will appear.

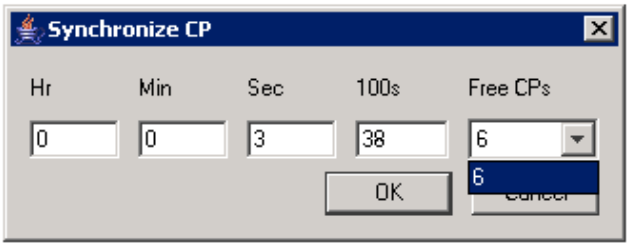


Figure X. Dialog for creating synchronizations.

The current position of the media will be filled in for you. Multitool will detect any free CP's around this point in time and display them in the *Free CPs* drop down list. (In this case CP5 and CP7 must already be synchronized since only CP6 is reported to be available). Select the CP that you want to synchronize and click *ok.* A TP will be generated and connected to CP6.

Creating a Synchronization from Scratch

To create synchronization is to create a CP connected to a point in time (TP).
We assume that there is currently no free CPs around the point in time where you want your synchronization. Place the blue "now line" where you want the new sync to be. From the menu; choose S*ync->New* and the "Synchronize CP" dialog will appear.

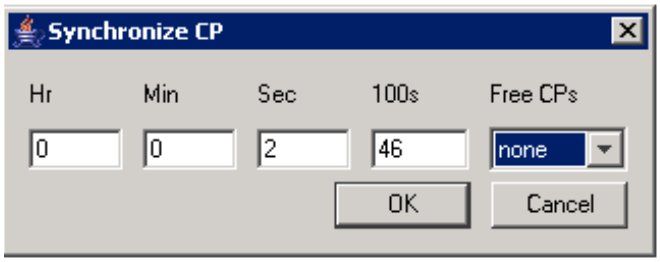


Figure X. Dialog for creating synchronizations.

The current position of the media will be filled in for you. Multitool try to auto detect any free CPs in the time range around the current time. If no free CPs are found the word "*none*" will be displayed in the *free CPs* list box. Just press OK and a question will pop up.

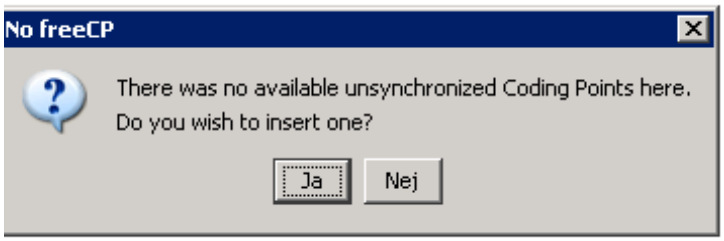


Figure X. Multitool asks if you want to create a new CP.

Press "*Ok*" here and a new CP will be created. The new CP will be attached to a TP with the time that was specified. Your new Synchronization is finished.

Creating a Synchronized Coding by Drag-Selecting

A synchronized coding has five elements to it; a start CP, an end CP, a start TP, and end TP and the coding. You can easily create this by left-clicking and dragging a selection in the TimeScale view, then select *Coding->New coding at current position.* The *Create new coding* dialog will appear. Type something in and press ok.

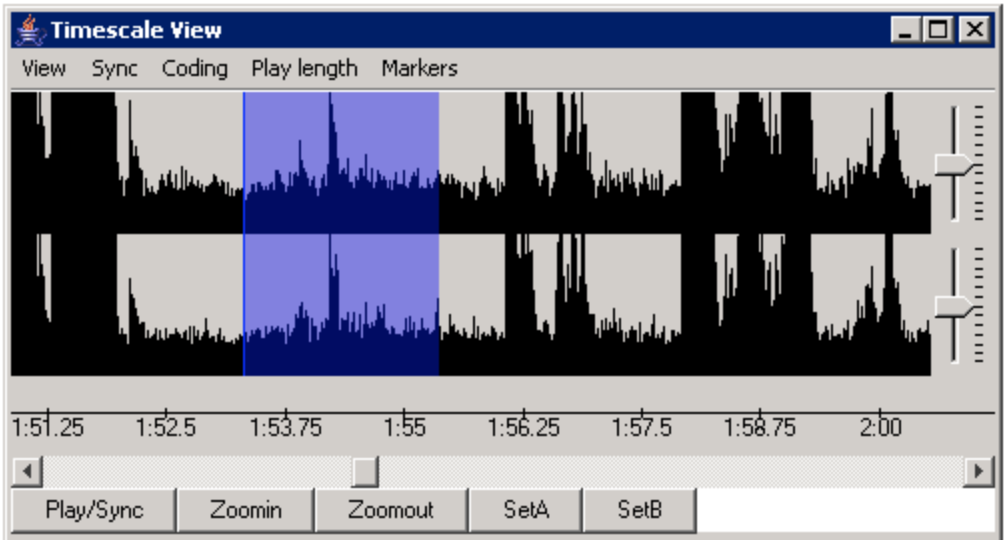


Figure X. A drag selection in transparent blue has been made in the TimeScale view. Note that the now line appears at the start of the selection.

You can have the selection traverse already existing synchronizations. Multitool will figure out and renumber coding point numbers automatically. Interval will also be filled in automatically in the *new coding* dialog. The field will be grayed out so you cannot change it. This is to avoid that the user will make an error.

There are some limitations on the drag selection mechanics that you should be aware of.

- You can only drag select from left to right (The now line will stay at the left side of the selection).

- You cannot drag selection outside of the current view. (If you need to make a longer selection and still be precise use the A/B points instead).
- Scrolling will make you loose the selection.

Creating a Synchronized Coding using A/B points

This works in the same way as creating codings with the drag-selection technique but you use the A/B points as start and end for the coding. Just place the now line where you want the coding to start and press the *SetA* button. A bright yellow line will appear. Now find the place where the coding should end by placing the now line there, then press the *SetB* button. A dark yellow line will appear. If you set the B point before the A point they will later automatically be switched. Now choose *Coding->New coding from A to B.* The *Create new coding* dialog will pop up. Fill in the dialog and press Ok.

Moving/Fine Tuning Synchronizations

You can move synchronizations back and forth in time in various ways.

Method 1
Double click on the synchronization in time scale bar. The synchronize CP dialog will appear. Type in a new value and press Ok.

Method 2
Single click on the synchronization in the time scale bar to select it. (Actually the clicking in the time scale bar selects the nearest sync, zoom in if you don't get it right). As you select a sync it turns red.

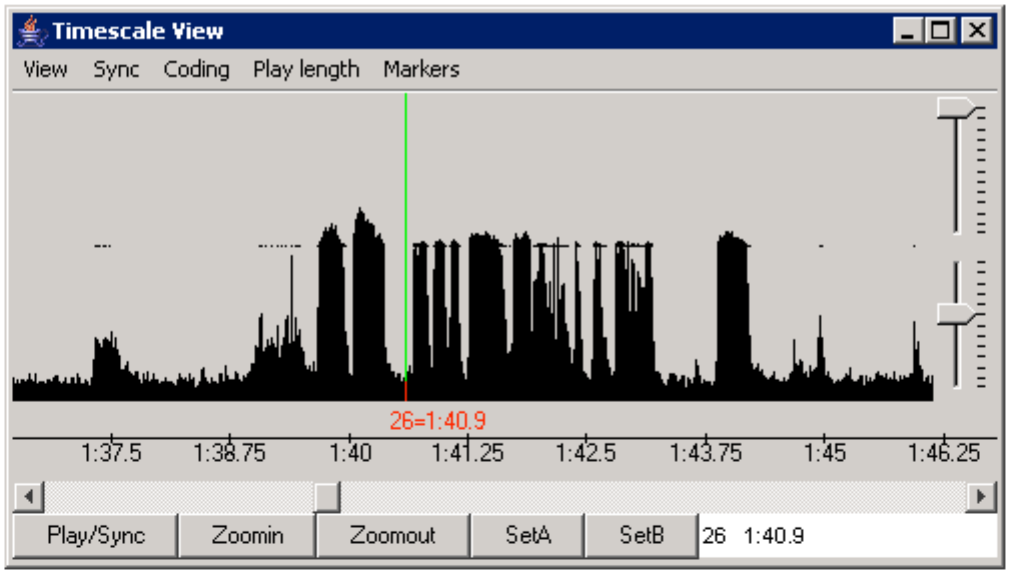


Figure X. A synchronization (CP 26 at TP 1 minute:40 seconds.9 tenths) has been selected as indicated in red.

With the sync selected you can move it using the *Sync* menus *earlier/later* commands. There are shortcut keys for these commands also. (Tip: The *earlier/later* commands operate on the now line if no sync is selected).

Caution: There is currently nothing that prevents you from moving a sync past another sync, possibly invalidating the temporal order of both coding points and time points.

Loop Playback

The loop playback features of Multitool are quite advanced. Some of them will be covered here and the rest of them will be covered in the Media View.

Observe that for the options below to work, playback must be started from the TimeScale view. If you start playback from, for example the partiture, the loop settings you have made in the TimeScale view will be ignored.

Looping a selection
Make a drag-selection and press space bar or the *Play/Sync* button.

Looping from currently selected synchronization to the next TP
From the *Play length* menu select *next tp* and start playback. (Tip: A/B points counts as next TPs too).

Looping from current position and X seconds forward
From the *Play length* menu select either *2 sec* or *5 sec* and start playback.

Looping between A to B points
Set the A/B points then click the *Play A-B* button in the Media view. (There is no Play A-B button in the TimeScale view but you can set A/B points here).

Reset looping
From the *Play length* menu select *continue.* Also make sure you don't have a selection by single clicking to position the now line.

It is also possible to create a coding while looping. You do this by making a drag-selection or setting up A/B point and starting looped playback. Then you fire up the *Create new coding* dialog. You can now type in information in the dialog while the section is being played back over and over. This is useful if you have a hard time hearing what a speaker is saying etc. (You cannot start looped playback after the dialog has been brought up).

Command Descriptions

Buttons

Sync/Play
Starts playback of media

Zoom in
Enlarges the amplitude waveform, horizontally.

Zoom out
Makes the amplitude waveform smaller, horizontally.

Set A
Drops the A-point marker at the current play position.

Set B
Drops the B-point marker at the current play position.

View

Zoom in
Equivalent to the button *Zoom in*.

Zoom out
Equivalent to the button *Zoom out*.

Align all
Equivalent to the button *Sync/Play.*

Close
Closes the window of the TimeScale view.

Sync

New
Creates new TimePoints and synchronizations.

Edit
Used to modify the selected synchronization.

Delete
Erases the currently selected synchronization.

Select next
Selects the next available synchronization, from the currently selected one.

Select previous
Selects the previous synchronization, from the currently selected one.

Deselect
Deselects any currently selected synchronization.

A little earlier / Earlier / Much earlier
Moves the currently selected synchronization back in time. If there is no synchronization currently selected it operates on the now line.
Warning: Be careful not to move a synchronization past another synchronization.

A little later / Later / Much later
Moves the currently selected synchronization forward in time. If there is no synchronization currently selected it operates on the now line.
Warning: Be careful not to move a synchronization past another synchronization.

Sync/Play
Equivalent to the *Sync/Play* button.

Coding

New coding at current position
Creates a new coding along with start and end synchronizations based on current drag-selection or current *now line* position, if no drag-selection exists.

New coding from A to B
Creates a new coding along with start and end synchronizations based on the location of A/B point markers. The A point will become the start sync and the B point will become the end sync. Both points must have been set prior to invoking this command.

Play_length

2 sec
Puts the TimeScale view in loop mode, looping 2 seconds from either current play position or from the currently selected synchronization.

5 sec
Puts the TimeScale view in loop mode, looping 5 seconds from either current play position or from the currently selected synchronization.

Continue
Disengages loop mode causing normal play.

Next TP
Puts the TimeScale view in loop mode, looping between the currently selected synchronization to the next.

Markers

Set A
Drops the A marker point at the *now line.*

Set B
Drops the B marker point at the *now line.*

Clear A/B points
Removes all A/B points, previously set.